REMARKS/ARGUMENTS

In view of the foregoing amendments and the following remarks, reconsideration of this application is requested. Claims 1-2, 4, 6-9, 11, 13-16, and 18-22 are now pending with claims 1, 6, 8, 13, 15, and 20 being independent. Claims 3, 5, 10, 12, and 17 have been canceled. Claims 1, 6, 8, 13, 15, and 20 have been amended.

The specification has been amended to provide the serial number of the coassigned application on page 9. The specification has been further amended to correct grammatical errors and minor informalities. The drawings have been amended to correct minor informalities. No new matter has been introduced.

Claims 1 and 8 have been amended in response to the Examiner's rejection under 35 U.S.C. § 102(e).

As amended, claim 1 describes a method for reducing total code size. The method includes determining a latency between a defining instruction and a using instruction. The method also includes inserting a NOP field into at least one of the defining and using instruction, wherein the NOP field is inserted at an end of the using instruction.

Amended claim 8 describes an apparatus having reduced total code size. The apparatus includes a processor containing at least one defining instruction followed by at least one using instruction. A latency exists between the at least one defining instruction and the at least one using instruction. At least one of the at least one defining and using instruction includes a NOP field, wherein the NOP field is inserted at an end of the using instruction.

Independent claims 1 and 8 stand rejected under 35 U.S.C. § 102(e) as anticipated by Suzuki (6,499,096). Applicants request reconsideration and withdrawal of these rejections for at least the reason that Suzuki does not describe or suggest the NOP field is inserted at an end of the using instruction.

Suzuki, in relevant part, describes in columns 1 and 2 compression of program code in a Very Large Instruction Word (VLIW) processor. As described in column 1, lines 35-57 the VLIW processor cannot simultaneously execute the two instructions "load mem (100) -> R1" and "add R1, R2->R3" because of the R1 dependency between the two instructions. Thus, these two instructions must be scheduled as shown in Code Example 1 by insertions of NOPs. To eliminate

the NOPs and compress the program code, format-specifying (FM) bits are assigned to lower order two bits of the 32-bit VLIW instruction containing the two instructions and the instructions are executed based on the value of FM bits as described in column 2, lines 18-25. Code example 3 shows the compressed VLIW instructions with the FM bits. In executing code example 3, the VLIW processor executes instructions as shown in Execution Sequence Example 1 given in column 2, lines 45-50.

Suzuki does not describe or suggest inserting the NOP field into the end of the using instruction (the Add). As shown in Code Example 3, Suzuki teaches compressing the code into a single 32 bit VLIW instruction code number 2 with FM bits assigned to lower order two bits of the 32 bits. The using instruction Add does not contain a NOP field at its end. Accordingly, for at least the reasons given above, Applicants respectfully submit that claims 1 and 8 are patentable over Suzuki.

The Examiner cited Mirapuri et al. (5,590,294) and Tsushima et al. (6,044,450) in the rejections for the remaining claims. For the sake of completeness, Applicants will explain how neither of these references describes or suggests the NOP field is inserted at an end of the using instruction as recited in claims 1 and 8.

Mirapuri, in relevant part, describes in column 2, lines 24-40 instruction pipelining for processors and efficient ways to restart stalled instruction pipelines. Latencies are present in a pipeline for a load or branch instruction-load instructions have a one cycle latency before the data is available for a subsequent instruction and branch instructions also have a one cycle latency while the instruction is fetched and the target address is determined. In conventional pipeline processors, latencies cause the processor to stall. One solution is to continue execution despite the interdependency and rely on software to avoid putting an instruction behind the instructions (i.e., load or branch) which need the information from the load or branch instructions before the information is ready. For example, the assembler can organize the instructions so that a useful instruction follows. If not possible to do so, a NOP instruction is inserted. Mirapuri does not describe or suggest that a NOP field is inserted at the end of the using instruction (the subsequent instruction) to specify NOPs before execution of the using instruction. In Mirapuri, the assembler determines if a NOP is needed after execution of a load or branch instruction.

Tsushima, in relevant part, describes in the Abstract and shows in Figures 2B, 11 and 12D small instructions that make up a VLIW long instruction added with a number of NOP instructions which succeed the small instruction, and these NOP instructions are deleted from the succeeding VLIW instruction. The small instructions have a NOP number field 12 as shown in Figure 2B specifying the number of NOPs that occur after the small instruction. The VLIW instructions are thus time-compressed. An instruction expanding unit has an instruction expanding circuit for each time compressed VLIW instruction. The instruction expanding circuit supplies after each small instruction, NOP instructions same in number as that designated by the NOP number field associated with each small instruction in the time-compressed VLIW instruction. Tsushima does not describe or suggest that a NOP field is inserted at the end of the using instruction to specify NOPs. Tsushima teaches that the NOPs occur after each small instruction (defining instruction) and accounts for the latency of the small instruction.

Claims 2, 4 and 9, 11 depend from independent claims 1 and 8, respectively. Accordingly, Applicants request reconsideration and withdrawal of the rejections for claims 2, 4, 9, and 11 for at least the reasons discussed above with respect to claims 1 and 8.

Claims 6 and 13 have been amended in response to the Examiner's rejection under 35 U.S.C. § 103(a).

Amended claim 6 describes a method for reducing total code size during branching. The method includes determining a variable latency after a branch instruction for initiating a branch from a first point to a second point in an instruction stream and inserting a NOP field into said branch instruction.

Amended claim 13 describes an apparatus for reducing total code size during branching. The apparatus includes a processor with at least one branch instruction for branching from a first point to a second point in an instruction stream, such that a variable latency exists in the branching between the first point and the second point. The at least one branch instruction includes a NOP field.

Independent claims 6 and 13 stand rejected under 35 U.S.C. § 103(a) as obvious over Mirapuri in view of Tsushima. Applicants request reconsideration and withdrawal of this rejection because neither Mirapuri nor Tsushima describe or suggest determining a variable latency after a branch instruction for initiating a branch from a first point to a second point in an

instruction stream. Mirapuri as mentioned above teaches that a branch instruction has a fixed one cycle latency while the instruction is fetched and the target address is determined. Tsushima teaches in Figure 9 instruction 38 is a BR LOOP instruction that performs a repetitive loop. Each time through the loop, the BR branch instruction has a fixed latency of zero cycles after execution as shown in Figures 11 and 12D "BR LOOP (0)" indicating that no NOPs occur after the BR branch instruction. Thus, neither Mirapuri nor Tsushima describe or suggest determining a variable latency after a branch instruction for initiating a branch from a first point to a second point in an instruction stream. Accordingly, Applicants request reconsideration and withdrawal of the rejection for at least the reasons discussed above with respect to claims 6 and 13

Claims 7 and 14 depend from independent claims 6 and 13, respectively. Accordingly, Applicants request reconsideration and withdrawal of the rejections for claims 7 and 14 for at least the reasons discussed above with respect to claims 6 and 13.

Claims 15 and 20 have been amended in response to the Examiner's rejection under 35 U.S.C. § 102(e).

Amended claim 15 describes a method for reducing total code size. The method includes locating at least one delayed effect instruction followed by NOPs within a code. The method also includes deleting the NOPs from the code and inserting a NOP field that specifies the deleted NOPs into a delaying instruction, wherein the deleted NOPs precede the delaying instruction.

Amended claim 20 describes an apparatus for reducing total code size. The apparatus includes a processor with a code containing at least one delayed effect instruction. The apparatus also includes a delaying instruction including a NOP field that specifies NOPs, wherein the NOPs precede the delaying instruction.

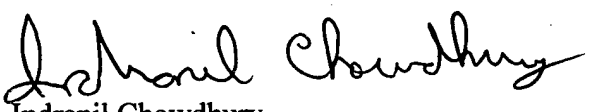
Independent claims 15 and 20 stand rejected under 35 U.S.C. § 102(e) as anticipated by Tsushima. Applicants request reconsideration and withdrawal of these rejections for at least the reason that Tsushima does not describe or suggest inserting a NOP field that specifies the deleted NOPs into a delaying instruction, wherein the deleted NOPs precede the delaying instruction. Tsushima, as mentioned above, teaches in the Abstract small instructions that make up a VLIW long instruction added with a number of NOP instructions which succeed the small instruction, and these NOP instructions are deleted from the succeeding VLIW instruction. The small

instructions have a NOP number field 12 as shown in Figure 2B specifying the number of NOPs that occur after the small instruction. Tsushima does not describe or suggest inserting a NOP field that specifies the deleted NOPs into a delaying instruction, wherein the deleted NOPs precede the delaying instruction. Accordingly, Applicants request reconsideration and withdrawal of the rejection for at least the reasons discussed above with respect to claims 15 and 20.

Claims 16, 18-19 and 21-22 depend from independent claims 15 and 20, respectively. Accordingly, Applicants request reconsideration and withdrawal of the rejections for claims 16, 18-19, and 21-22 for at least the reasons discussed above with respect to claims 15 and 20.

In view of these remarks and amendments, Applicants submit that this application is now in condition for allowance and the Examiner's prompt action in accordance therewith is respectfully requested. The Commissioner is authorized to charge any additional fees and/or credit any overpayment to Deposit Account 20-0668 of Texas Instruments Incorporated.

Respectfully submitted,



Indranil Chowdhury
Attorney for Applicants
Reg. No. 47,490

Robert D. Marshall, Jr.
Texas Instruments Incorporated
P.O. Box 655474, MS 3999
Dallas, TX 75265
(972) 917-5290

Attachments

OPCODE																													
OFFSET	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME		creg		z						scst12							ucst3			0	0	0	0	1	0	0	1	0	0	0	s	p
WIDTH		3		1						12							3			1				10							1	1

FIG. 4a

PIPELINE STAGE	E1	PS	PW	PR	DP	DC	E1
READ	src2						
WRITTEN	PC						
BRANCH TAKEN			N				X
UNIT IN USE	.S						

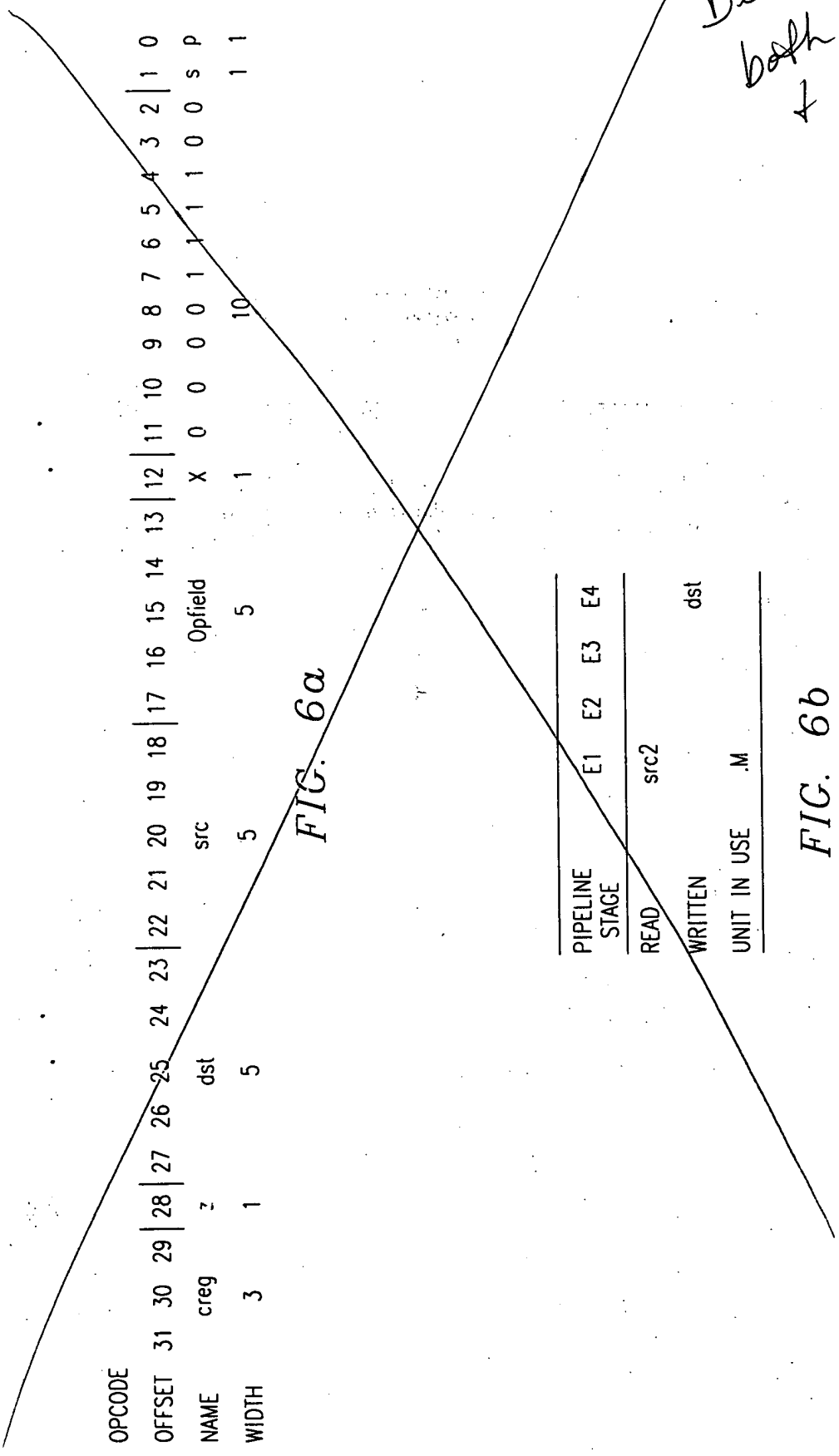
FIG. 4b

PIPELINE STAGE	E1	PS	PW	PR	DP	DC	E1
READ	src2						
WRITTEN	PC						
BRANCH TAKEN							X
UNIT IN USE	.S2						

FIG. 5b

OPCODE																													
OFFSET	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAME		creg		z	0	0	0	0	1			src2			0	0		ucst3		x	0	0	1	1	0	1	1	0	0	0	1	p
WIDTH		3		1			5					5			2			3		1				10							1	1

FIG. 5a

Appl. No. 09/702,484
Amdt. Dated Feb. 27, 2004
Response to Office Action of October 1, 2003
Annotated Sheet 1 Showing Changes

Delete both 6a + 6b